# Project Report

# on

# HANDWRITTEN DIGIT RECOGNITION USING CNN

(A Project Report submitted in partial fulfillment of the requirements of Bachelor of Technology in Information Technology of the West Bengal University of Technology, West Bengal)

Submitted by

Biswajit Mirdda (10200217058)

Debalina Mitra (10200217056)

Rabi Kumar Shaw (10200217041)

# Under the guidance of

Mrs. Manisha Barman

Professor / Associate prof. /Asstant Prof./Lecturer,

Dept. of Information Technology

# 

# Kalyani Government Engineering College

(Affiliated to West Bengal University of Technology)

## Kalyani - 741235, Nadia, WB

2020 - 2021

**Phone: 25826680 (PBX)**

**Fax : 25821309**



**Certificate of Approval**

This is to certify that …………………………………has done final year project work entitled…………………………………………under my direct supervision and he/she has fulfilled all the requirements of relating to the Final Year Project. It is also certified that this project work being submitted, fulfills the norms of academic standard for B. Tech Degree in Information Technology of The West Bengal University of Technology and it has not been submitted for any degree whatsoever by him/her or anyone else previously.

………………………………………… ……………………………………………..

Head Supervisor

Department of Information Technology Department of Information Technology

Kalyani Government Engineering College Kalyani Government Engineering College

………………………………………… ………………………………………………

Project Coordinator Examiner

Department of Information Technology

Kalyani Government Engineering College

### ACKNOWLEDGEMENTS

Success of any project depends largely on the encouragement and guidelines of many others. We take this sincere opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project work.

We would like to express our sincere gratitude to **Mrs. Manisha Barman, Assistant Professor**, Department: Information Technology, Kalyani Government Engineering College. We always feel motivated and encouraged every time by her valuable advice and constant inspiration; without her encouragement and guidance this project would not have materialized.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skills that will always remain as our asset and to all non-teaching staff for the cordial support they offered.

Place: Kalyani Government Engineering College

Date: 01.04.2021

1. BISWAJIT MIRDDA - 10200217058
2. DEBALINA MITRA -10200217056
3. RABI KUMAR SHAW - 10200217041

**ABSTRACT**

To make machines more intelligent, the developers are diving into the machine learning and deep learning architecture. In the real world, a human learns to perform a task by practicing and repeating it again and again so that it memorizes how to perform those tasks. In this way, the biological neurons in his brain automatically get triggered and they can quickly perform the task they have learned. Deep learning is also very similar to this. It uses different types of neural network architectures for different types of problems. **For example –** digit recognition, object recognition, image and sound classification, object detection, image segmentation, etc.

Digit Recognition is a noteworthy and important issue. As the manually written digits are not of a similar shape, size, thickness, position and direction, in this manner, various difficulties must be considered to determine the issue of handwritten digit recognition. The uniqueness and assortment in the composition styles of various individuals additionally influence the example and presence of the digits. It is the strategy for perceiving and arranging transcribed digits. The task of handwritten digit recognition, using a classifier, has great importance and use such as – online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand (for example ‐ tax forms) and so on

In this project, we are going to implement a handwritten digit recognition algorithm using the MNIST dataset. We will be using a special type of deep neural network that is **Convolutional Neural Network (CNN)**. In the end, we are going to build a GUI model by which we can open paint and draw the digit and then recognize it straight away.

The aim of this project is to implement a classification algorithm to recognize the handwritten digits. The after effects of probably the most broadly utilized Machine Learning Algorithms with Deep Learning calculation like multilayer CNN utilizing Keras with Tensorflow. Utilizing this, the accuracy we got is of 98.70% utsing CNN. However this accuracy can be varied in different algorithm.

**TABLE OF CONTENTS**

**CHAPTER 1 INTRODUCTION 01**

* 1. **Motivation 02**
  2. **Background 03**
  3. **Summary of present work 04**
  4. **Hardware/Software used 05**

**CHAPTER 2 TOOLS 06**

**2.1 MNIST Dataset 06**

**2.2 Keras Library 07**

**2.3 Tensorflow Library 07**

**2.4 Numpy 07**

**2.5 Pillow 07**

**2.6 Matplotlib 07**

**2.7 Tkinter 08**

**CHAPTER 3 METHODOLOGY 08**

**3.1 Getting Familiar with the Tools 08**

**3.2 Convolutional Neural Network 08**

**CHAPTER 4 ARCHITECTURE 12**

**4.1 Block Diagram 12**

**4.2 Sequence Diagram 13**

**4.3 Dataflow Diagram 13**

**CHAPTER 5 IMPLEMENTATION 14**

**5.1 Import the libraries and load the dataset 14**

**5.2 Data Pre-Processing 14**

**5.3 Segmentation 14**

**5.4 Feature Extraction 14**

**5.5 Creating the model 15**

**5.6 Creating the GUI 16**

**5.7 Result 17**

**LIMITATIONS 18**

**CONCLUSIONS AND FUTURE WORK 19**

**REFERENCES 20**

1. **INTRODUCTION**

It is very easy task for the human brain to process the images and analyse them. When the normal eye sees a certain image, the brain can easily segment it and recognize its different segmented attributes. The brain automatically goes through that process, which involves not only the analysis of this images, but also the comparison of their different characteristics and attributes, with what it already learnt in order to be able to recognize these images. There is a field in computer science that tries to do the same thing for machines, which is Image Processing.

Image processing is the field that concerns with analysing the images so as to extract out some useful information from them. This method takes images and converts them into a digital form by some process, and that form is readable by computers or machines, it applies certain algorithms on them, and results in a better quality images or with some of their characteristics that could be used in order to extract out some important information from them.

Image processing is applied in various fields, especially nowadays, and various softwares have been developed that use this concept. Now we have self-driven cars which can recognize other cars and human beings to avoid accidents. Also, some social media applications, like Facebook uses face recognization technique which is also a part of image processing. Furthermore, some softwares use it in order to recognise the characters in some images, which is the concept of digit recognition, that we will be discussing and implementing in throughout this project.

In this project, we will try on building a mechanism that will recognize handwritten digits. We will be reading images containing handwritten digits extracted from the MNIST dataset and will try to recognize which digit is represented by that image. For that we will use some machine learning technique. This approach is based on matrices manipulations, as it reads the images as matrices in which each element is a pixel. It 2 overlaps the image with all the images in the reference set and find the correlation between them in order to be able to recognize which digit it represents.

After building the machine learning model we will try to make a Graphical User Interface (GUI) by which an user can predict their own handwritten digit. They will draw an image and our system will capture that image, process them, and then will try to recognize which digit it represents.

The goal of this project is to apply and manipulate the basic image correlation techniques and apply a neural network architecture to build program and keep polishing and enhancing in order to investigate to which direction it can get improved. This would allow us to understand how good we can do, in terms of accuracy and performance, using simple and basic techniques as well as some complex computation of machine learning.

01

* 1. **MOTIVATION**
* Before computers existed all the information was stored in written form in the papers. This is very old and ineffiecient form of storage as the paper information cannot be stored for a long time and it can get lost or be destroyed.
* On the other hand information on computer is stored safely for a long time and multiple copies of same information can be made easily.
* Thus after inventing the computers a lot of money was wasted in manual human labor for converting this paper information into digital information.
* Instead of this, machine learning can be used to recognize and convert this information and data from paper documents into digital format without wasting money on human labor.
* A numerous attempt of research work in the field of machine learning and data mining has been contrived to achieve efficient approaches for approximation of recognition from paper information or data.
* In handwritten digits there are different shapes, sizes and directions present. As the handwriting of every individual are not same.
* Handwritten digit recognition is one of the practically important part in pattern recognition applications. The applications of digit recognition includes in postal mail sorting, bank check processing, form data entry, etc.
* So by applying this model, computers can able to recognize digits automatically without being explicitly programmed for individual.
* Our project is a approach of this recognition technique in the machine learning field.
* After building the model we will make a GUI by which we can draw digits on paint and recognize it accordingly by clicking on the button present on it.

**1.2 BACKGROUND**

02

Before going into the implementation we have to know some basic and main concepts of digit recognition and machine learning.

In this section, we will discuss the main concepts like machine learning, image processing, neural network etc that this project is concerned with.

**1.2.1 Image Processing:**

Image processing is a very large and important field within computer science and technology which deals mainly with analysing images and trying to extract some information or attributes out of them. The image to be processed is imported first, then analysed using some computational methods, which, by the end, results either in an image with a better quality or some of the characteristics of this image depending on the purpose of this analysis. This is a very wide field within computer science, which also has several other subfields, out of which Digit Recognition is one of them, that we will be mainly dealing with throughout this project.

**1.2.2 Handwritten Digit Recognition – History:**

It is easy for the normal eye to recognize a digit when written on any paper; however, computers cannot identify the characters from an image or scanned document. In order to make this possible, a lot of research has been done, which resulted in the implementation of various algorithms that made this possible. One of the fields that specialize in character recognition under the light of Image Processing is Handwritten Digit Recognition.

**1.2.3 Machine Learning:**

Machine learning simply means that by which a machine can learn. It is a subfield of artificial intelligence based on the idea that machines can learn from data and make dicission like human do.

Machine learning is a field that concerns making programs learn and know how to behave in different situations using the data. There are two different types of machine learning algorithm which are Supervised learning and another one is Unsupervised learning.

There are also many sub fields of these two learning algorithm.

One of its applications is Handwritten Digit Recognition that we are going to develop in this project.

**1.2.4 Artifical Neural Network:**

An Artificial Neural Network (ANN) is a system that tries to mimic the human’s biological neural network in the brain. It is an algorithm used for machine learning, which means it uses data to learn how to respond to different input data. The ANN can be seen as a device, which takes one or more inputs and gives one output. Inside the device, there exist several interconnected nodes. The input is brought into the program, which passes through the several layers and nodes of the ANN and gives an output using a computaional function.

03

Artificial Neural Network is used for Handwritten Digit Recognition and have proved a very high accuracy rate. In this case, the ANN would “recognize a digit based on its physical features such as shape, size, symmetry, closed or open areas, and number of pixels etc”. The high accuracy of this kind of algorithms is mainly thanks to its ability of learning from the training set, which would contain characters with similar features. A type of ANN is Convolutional Neural Network (CNN) which we have used to build our model.

**1.2.5 Graphical User Interface:**

A GUI (Graphical User Interface) a system by which an user can communicate with the computer software. A GUI shows object that convey information and display result of the actions taken by the user.

**1.3 SUMMARY OF PRESENT WORK**

In order to be able to implement our recognizer and test its performance, it is necessary to have a suitable dataset which contains a large number of handwritten digits. This dataset should be able to allow us to discover the challenges and limitation of the image correlation technique and push us to look for ways and methods to enhance it and assess its accuracy. We have opted for the MNIST dataset to be used for testing our program since it has proved a great reliability and importance in the field.

Each image is a 28 by 28 pixel square (784 pixels total). A standard split of the dataset is used to evaluate and compare models, where 60,000 images are used to train a model and a separate set of 10,000 images are used to test it.

In this project, we have created a CNN model which can recognize the digits and give the accuracy rate. We have taken the MNIST dataset which contains total of seventy thousand of digits of different shapes and size. Then we have trained the dataset using our algorithm and then tested the model. Afterwords we have checked the accuracy rate and calculated the error. In the output we have shown the the predicted digit using matplotlib with accuracy rate.

After developing the model, We have created a Graphical User Interface (GUI) and have loaded our model. On the window of the interface there are two different buttons. Open Paint button is used to open the paint application in our computer. After opening we can draw the digit on paint by using different brush. After completing the drawing, We can predict that image by pressing another button Live Prediction on the interface, and will show the predicted result along with the accuracy percentage. So basically by using the GUI one can predict the digit drawn by himself/herself without understanding the internal implementation or code.

**1.4 Software/Hardware Used:**

04

04

**1.4.1 Software Used:**

**A. Python 3.7:**

Python is a broadly utilized universally and is a high-level programming language. It was primarily introduced for prominence on code, and its language structure enables software engineers to express ideas in fewer lines of code. Python is a programming language that gives you a chance to work rapidly and coordinate frameworks more effectively.

**B. Anaconda3.5.3.1:**

Anaconda is a free and open-source application of the Python and R programming for logical analysing like information science, AI applications, large-scale information preparing, prescient investigation, and so on. Anaconda accompanies in excess of 1,400 packages just as the Conda package and virtual environment director, called Anaconda Navigator, so it takes out the need to figure out how to introduce every library freely. Anaconda Navigator is a graphical UI (GUI) incorporated into Anaconda appropriation that enables clients to dispatch applications and oversee conda packages, conditions and channels without utilizing commandline directions.

In the Anaconda package, We have used Jupyter notebook to write our code and develop our model.Jupyter notebook is very useful to building python models and it could be managed easily.

**C. Paint:**

This is a software application of Windows Operating System. In this Paint application, we will draw the digit with the help of different brush to recognize the digit accordingly.

In order to visualize how our model can perform on input images, we need to draw the image on paint application.

**2.TOOLS**

05

This project’s main objective is to be able to read the images containing the handwritten digits and be able to identify those digits using basic image correlation techniques. These images are normally represented and read as matrices, in which every element portrays a pixel. The image correlation technique takes these matrices and compares them using some algorithms so as to identify the match that represents the digit we are trying to figure out. This project will be mainly using matrices and heavy numerical computations, that is why it is very important to consider the tools that would provide us with a suitable environment for performing these computations.

**2.1. MNIST Dataset:**

The MNIST dataset, which stands for the Modified National Institute of Standards and Technology database, is a very large dataset containing seventy thousands of handwritten digits. This dataset was created by mixing different sets inside the original National Institute of Standards and Technology (NIST) sets. So as to have a training set which contain svarious types and shapes of handwritten digits, as the NIST set was divided into those written by high school students and others written by the Census Bureau workers . The MNIST dataset has been the main source of data of so many research done in recognizing handwritten digits. This allowed the development and improvements of many different algorithms with a very high performance, such as machine learning classifiers.

In order to be able to implement our recognizer and test its performance, it is necessary to have a suitable dataset which contains a large number of handwritten digits. This dataset should be able to allow us to discover the challenges and limitation of the image correlation technique and push us to find the ways and different implementation to improve it and increase its accuracy. We have choosen this dataset to be used for training our program/model since it has proved a great reliability and importance in the field.

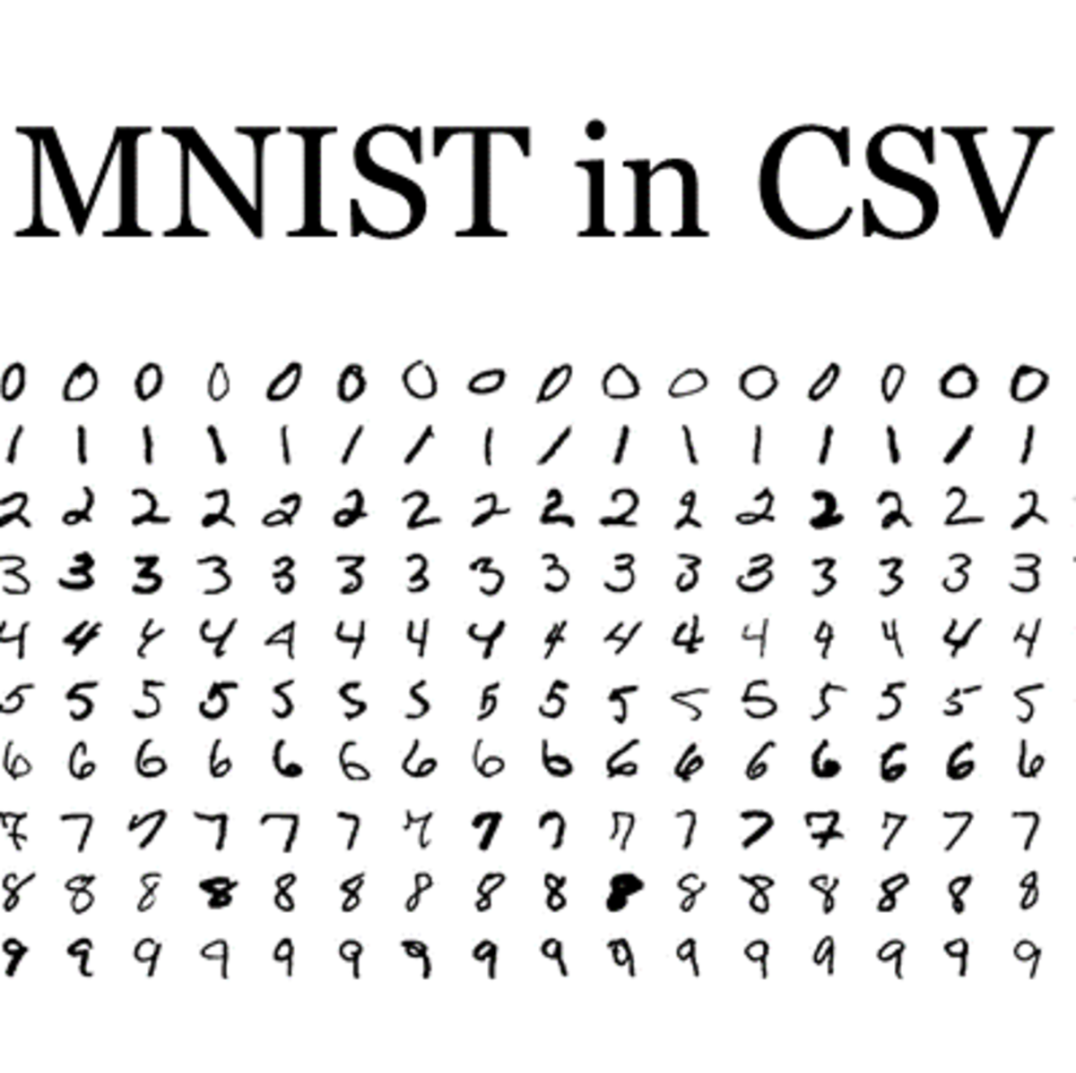


Figure 2.1: MNIST Dataset

06

**2.2 Keras Library:**

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides. By using keras we can load our file or image from our computer folders.

**2.3 Tensorflow Library:**

TensorFlow is an amazing information stream in machine learning library made by the Brain Team of Google and made open source in 2015.It is intended to ease the use and broadly relevant to both numeric and neural system issues in machine learning field.

Fundamentally, TensorFlow is a lowlevel tool for doing entangled math and it targets specialists who recognize what they're doing to construct exploratory learning structures, to play around with them and to transform them into running programs.

For the most, it can be considered as a programming framework in which one can entitle to calculations as graphs. Nodes in the graph speak the math activities, and the edges contain the multi-dimensional information clusters (tensors) related between them.

**2.4 Numpy:**

NumPy is a library for the Python programming language. It supports a wide, multi-dimensional arrays and matrices, along with a lot of collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin.

Using NumPy in Python gives comfortale coding experience as it allows the user to write fast programs as long as most operations work on arrays or matrices instead of scalars.

By using numpy module, we can handle and manipulate a large number of array data format files.

**2.5 Pillow:**

Pillow is also a module of python library which is called PIL (Python Image Library). PIL is one of the important Library for image processing technique in Python.

Pillow module gives more functionalities, runs on all major operating system and support for python 3. We can do almost everything on digital images using the pillow module. Apart from basic image processing functionality, we can also do filtering the images using built-in convolution layers, and color space conversions.

**2.6 Matplotlib:**

Matplotlib is a very useful library in python to show or to visualise the digital images. It is used to visulise the image as well as to show the graphical plotting on the axis in the programming output.

07

**2.7 Tkinter:**

Tkinter is a module supported in python used for creating GUI (Graphical User Interface). It is the most used and liable toolkit to make GUI. It provides fast and easy uses to implement a GUI . We can install Tkinter module by running the command “pip install Tkinter”. We can give the different width and height also in the Tkinter GUI.

**3. METHODOLOGY**

**3.1 Getting Familiar with the Tools:**

Before going into the main program we first need to get familier with the softwares and their necessary libraries and their usages. So after the installation we need to know the behaviour and limitations of the tools we used.

The first step we had to go through while working on this project was getting familiar with the tools used, i.e., Python and the MNIST dataset. After setting up the required environment for Python to work perfectly and downloading the dataset, we have started experimenting the details of the language in order to get familiar with them and know how to use them comfotably in the future implementation.

After downloading the required modules I have started learning and experimenting different tools like keras, tensorflow, numpy, sklearn and started working with MNIST dataset.

The MNIST dataset, which was used to create our training and testing part, contains total seventy thousand of handwritten digits, represented as matrices. It has been used in the development of various research work and programs as well as in many projects with the same goal as ours. After downloading the file which contains the handwritten digits, I have loaded it on Jupyter Python Notebook in order to visualize the images and figure out how to use and manipulate them. For that we need to understand the hidden pattern like shapes, sizes, directions and we need to know how to work with them.

After downloading the required libraries we need to know the how to import them into the code and what are their different functions and their uses. So that we should be able to use all these things comfortably in our actual model implementation.

* 1. **Convolutinal Neural Network:**

After getting familier with the necessary toolkits, now is the time to understand our actual concept of this project.

The main agenda for this field is to enable our machines to view the outside world as humans do, realize it in a similar manner and even use the learning for a variety of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Pattern Recognition System, Recommendation Systems, Natural Language Processing, etc. The progress in Computer Vision with Deep Learning has been constructed and more accurate with time, primarily over one important algorithm — a **Convolutional Neural Network (CNN)**.

In the past few years, Deep learning has proved as a very powerful tool in computer science field because of it’s ability to learn and manipulate a large number of data. The interest to find the hidden layers and use it has crossed the past techniques, more especially in pattern recognition field.

08

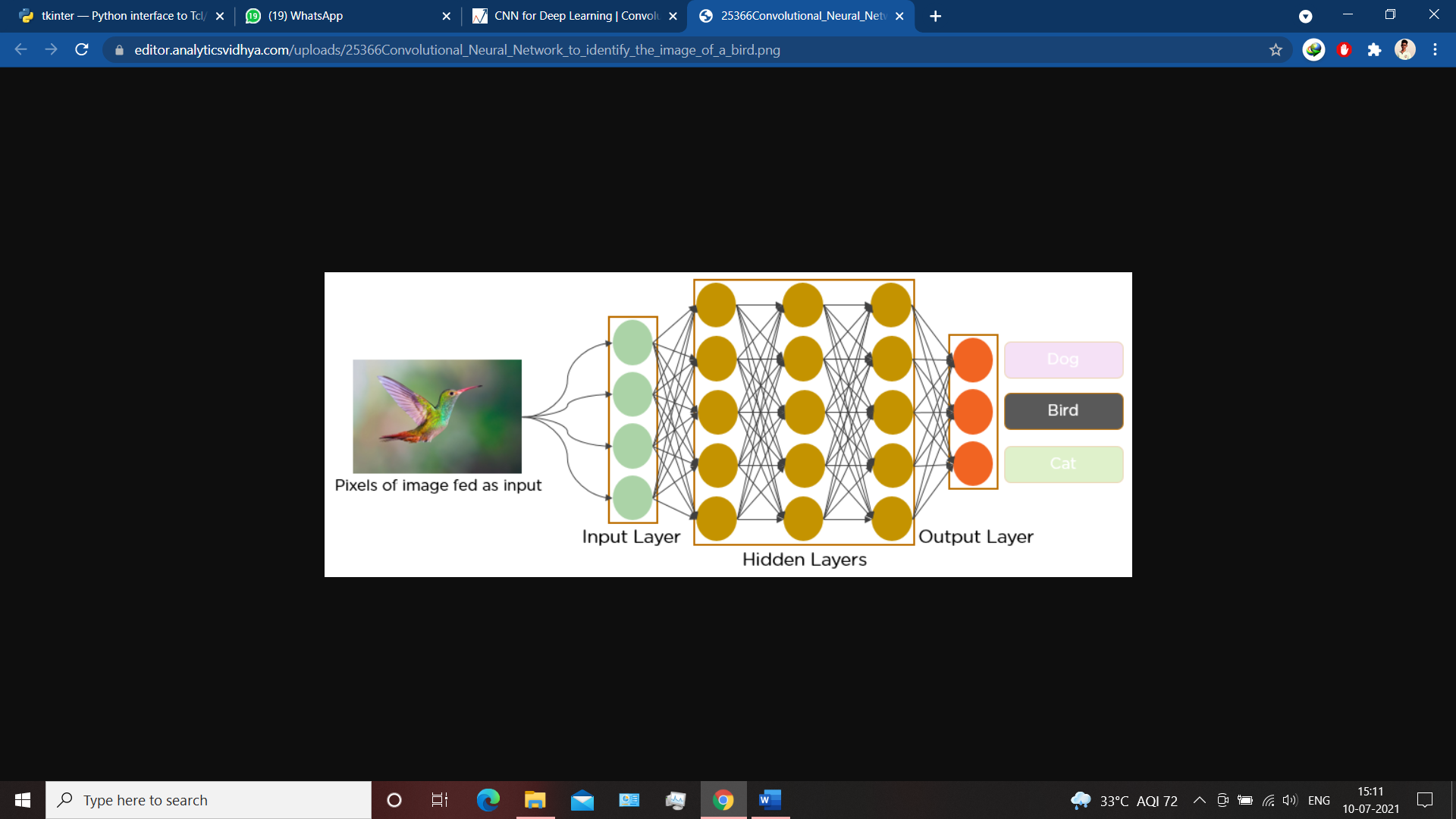


Fig 3.1 A visualization of Convolutional Neural Network

A **Convolutional Neural Network (ConvNet/CNN)** is a class of Deep Neural Network which can take an input image, assign importance (learnable weights and biases) to various aspects/objects like shapes, sizes, directions of the image and be able to differentiate one from another. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics of the digits. The above figure is a visual representation of Convolutial Neural Network.

**How CNN Works:**

Before going into the working of CNN lets understand some basic concepts like what is an image and how it is represented in computer. Image can represented in two types- one is RGB (Red, Green, Blue) and the other is gray image. A coloured or RGB image is a image which contains a matrix of pixel values in three different channels. On the other hand, a grayscale image is the same but with one single channel. Take a look at this image to have a better understand.

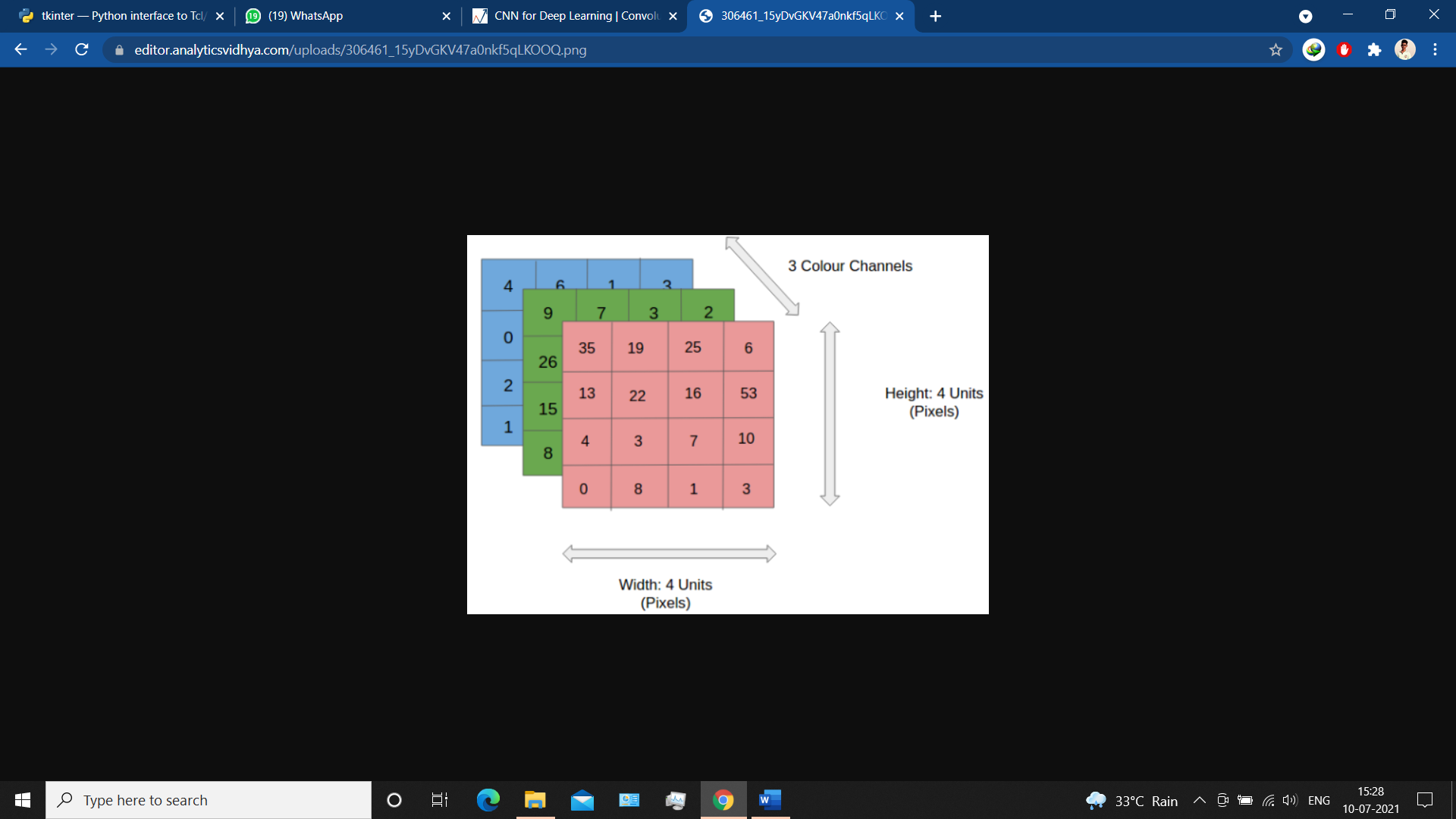


Fig 3.2 Visualization of an RGB image in array format

Now lets know what are the different layers of an CNN algorithm and how does they work in order to analyse an inage. And after analysing how it can recognize an image differently.

09

There are four different layers in CNN. Those are-

1. Convolution Layer
2. ReLU
3. Pooling
4. Fully Connected Layer

We will discuss every layer one by one. But before that lets visualize how the layers are connected to each other and how the input passed from one layer to other.

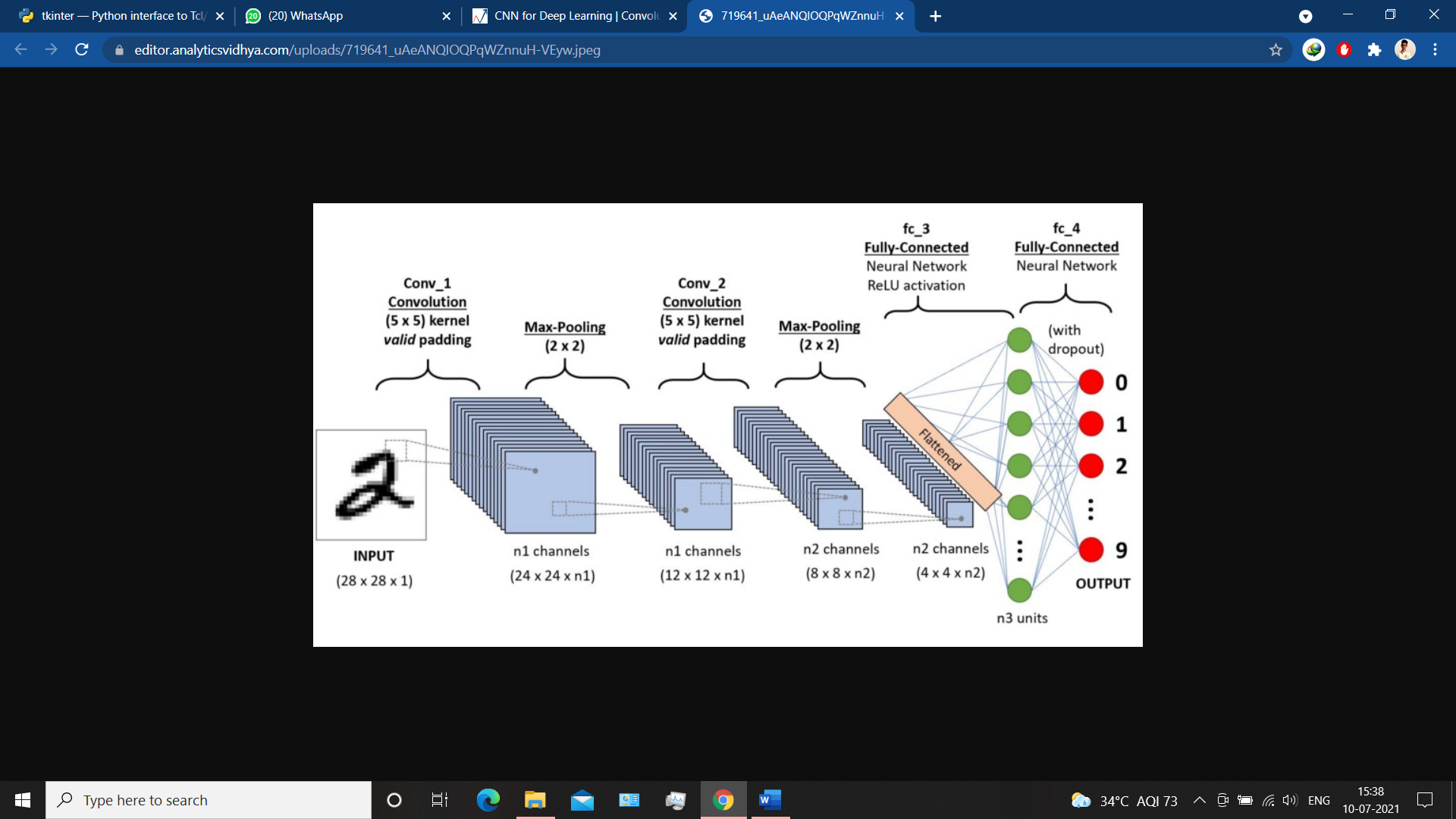


Fig 3.3: Representation of different layers of CNN

**1. Convolution Layer:**

The first layer of CNN is the convolutional layer which takes an image as input. After taking the input it converts that image into grayscale inorder to process and analyse them further. Below are the steps that happen.

1. First line up the feature and the image.
2. Multiply those image pixel by the corresponding feature map.
3. Add them up
4. Divided by number of pixels in the feature.
5. Move the filter through out the image.
6. We have to apply first 5 steps for all the features, we used here.

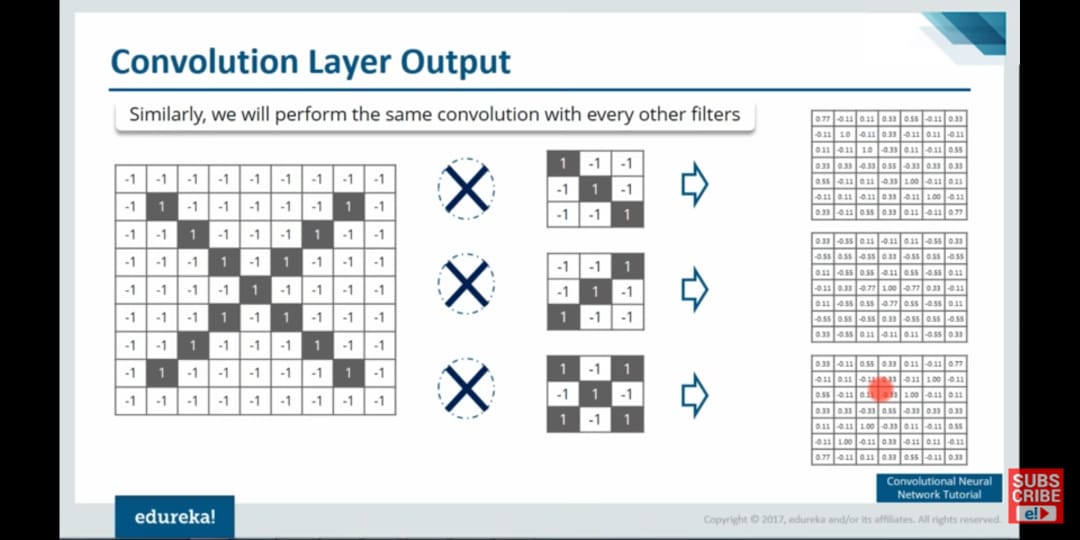


Figure 3.4: Convolutional Layer output

10

**2. Relu Layer:**

ReLU is the second layer of the CNN architecture, abbreviation of Rectified Linear Unit. It is nothing but a transform activation function which activates a node when the input is above a certain value. When the input is zero, Output will be zero, but when input goes beyond to a particular threshold, it shows a linear relationship with dependant variables.

**3.Pooling Layer:**

The next layer of our CNN model is Pooling layer. Pooling Layer is basically used to reduce the size of an image. There are two types of pooling available- one is average pooling and other is Max pooling. We have used Max pooling in our implementation.

The Steps that involved in pooling are:

1. Pick a window of size 2.
2. Move the window Throughout the matrix.
3. Take the maximum value from each window so that we can shrink the image.

**4.Fully Connected Layer:**

The last layer of our CNN model is Fully Connected Layer. After stacking convolution,ReLU and pooling layers we make the input image shrinked and then it comes to Fully Connected Layer. This is the final layer where the actual classsification happens. In this layer we use “softmax” function which is used in multi-class classifier problem.

Here we take all the filtered and shrinked images and put it into a single layer.



Figure 3.5: Matrix Representation After the layer

After taking it into one single layer there will be some value as high(1) and some will be as low(<1) for different images. From this pattern our model will learn to classify the image. Below is the image that shown the output after the fully connected layer.

11

09

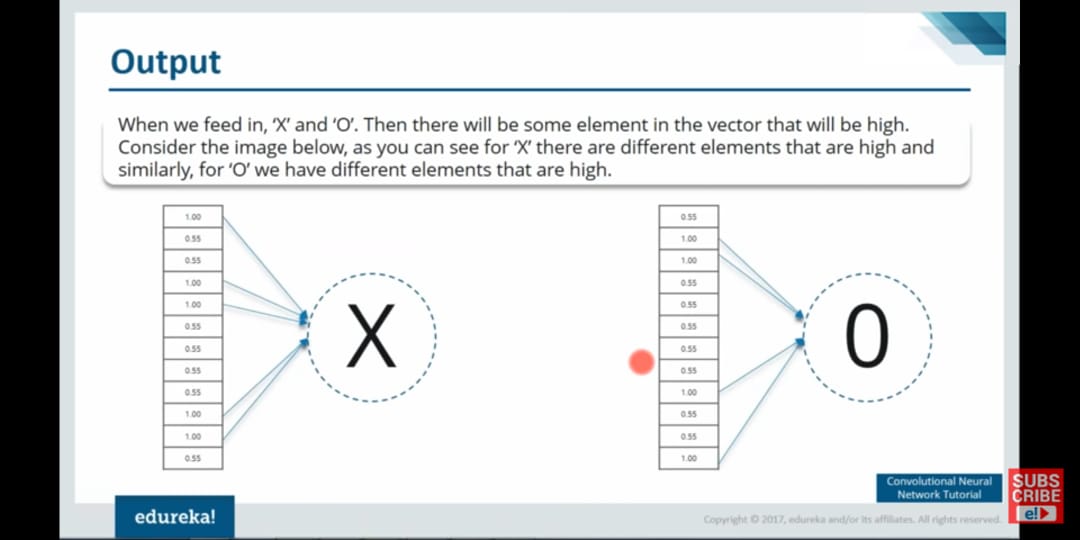


Figure 3.3: Output After The Layer

**4. ARCHITECTURE**

The reason behind this document is to look into the design possibilities of the proposed system, such as architecture design, block diagram, sequence diagram, data flow diagram and user interface design of the system in order to define the steps such as pre-processing, feature extraction, segmentation, classification and recognition of digits.

**4.1 Block Diagram:**

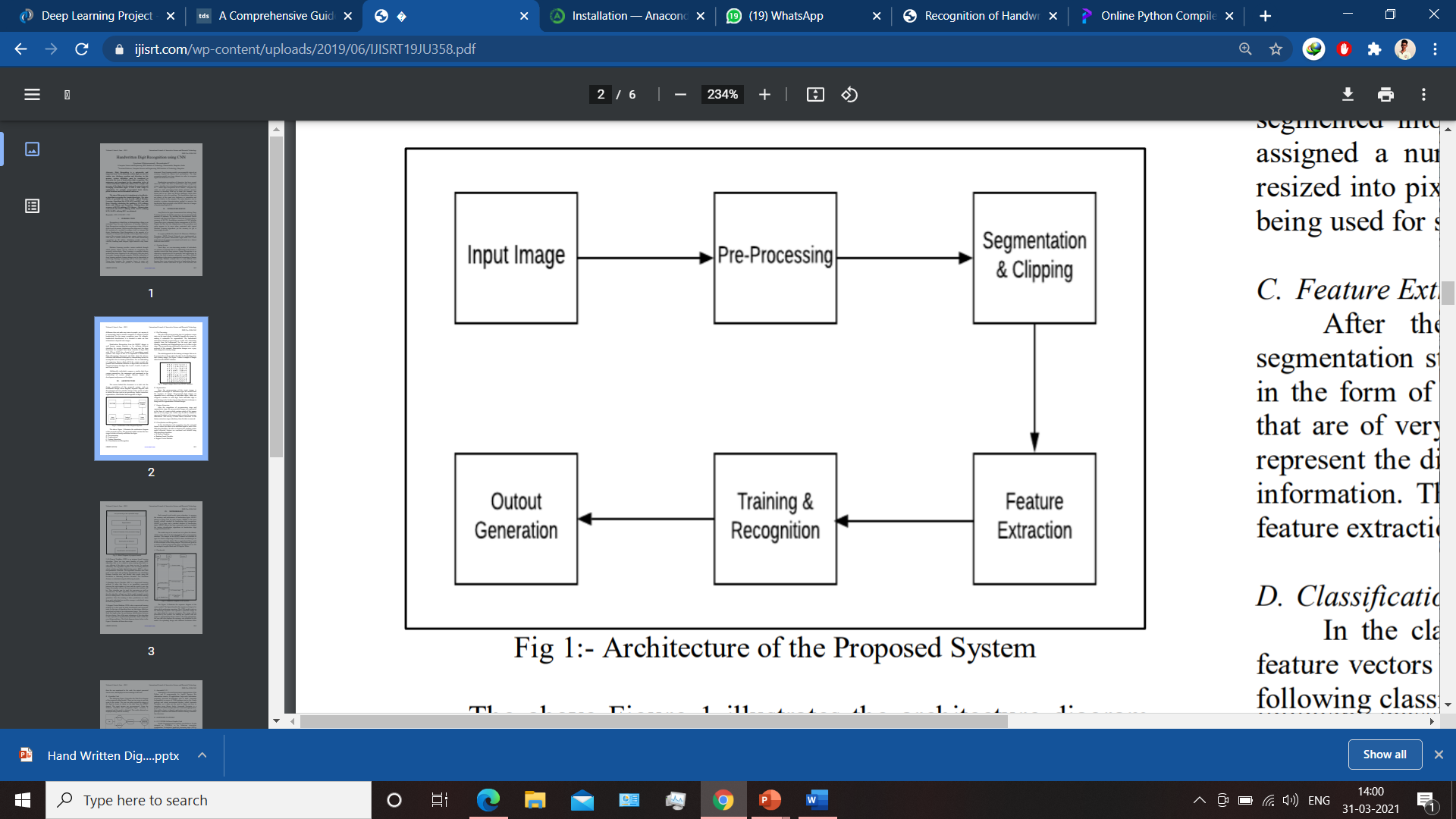


Figure 4.1: Block Diagram of proposed system

The above Figure 4.1 demonstraits the block diagram of the proposed system. The proposed model contains the four stages in order to classify and detect the digits:

A. Pre-processing

B. Segmentation

C. Feature Extraction

D. Classification and Recognition

We will discuss about each stage while implementing the model.

12

**4.2 Sequence Diagram:**

The Figure 4.2 illustrates the sequence diagram of the system model. The figure describes the sequence of steps to be taken while performing execution. The CNN model works in the following sequence. User uploads a particular image of any digit which he wants to recognize. The image will be processed by the system. On running the system code the output is generated that shows which is the digit uploaded by the user and also displays the accuracy rate predicted by the model.

Figure 4.2: Sequence Diagram

**4.3 DataflowDiagram:**

There are two different ways to provide input to the system. The user can either upload the image of the digit he wants to detect or the data from the MNIST dataset. The input images have to be pre-processed before predicting. Using the CNN classifiers the recognized digits’ accuracy is obtained and the result is shown. After that the results obtained are displayed along with the accuracy. Below figure is the dataflow diagram of our system.

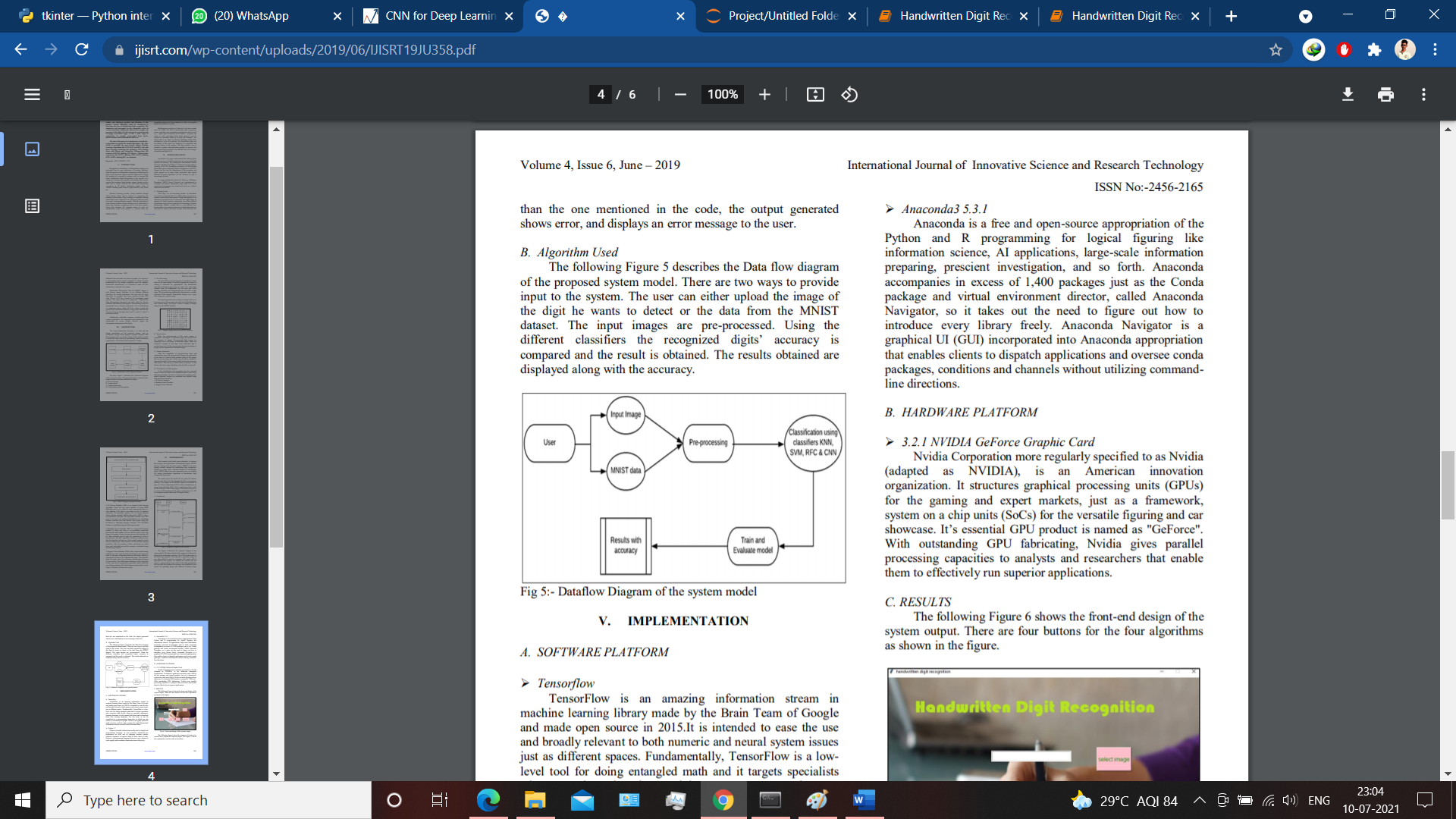


Figure 4.3: DataFlow Diagram

13

**5.IMPLEMENTATION**

In this project we aim at developing a architecture that would be able to recognize handwritten digits from the MNIST dataset. We have opted for the Convolutional Neural Network architecture for this project and our goal is to use the basic and simple as well as some complex computing and see how good can we make the accuracy rate.

After implementing the model, we have tried to spot the problems in the mechanism and find the limitations of the technique in order to improve it, and finally we made a GUI by which an user can predict a digit drawn by himself. We kept doing the program by increasing the layers, each time trying to improve the previous version, which enabled us to keep improving the program and reach a higher accuracy and performance.

### 5.1 Import the libraries and load the dataset:

### To develop the model we need some library and need to know their behaviour and limitaions. First, we are going to import all the modules that we are going to need for training our model. The Tensorflow library already contains some datasets and MNIST is one of them. So we can easily import the dataset and start working with it. The **mnist.load\_data()** method returns us the training data, its labels and also the testing data and its labels. Then we can split the dataset into training and testing part.

**5.2 Pre-processing:**

After importing the necessary libraries we need to pre process the image. The role of the pre-processing step is it performs several tasks on the input image. It basically upgrades the image data by making it logical for segmentation. The primary and main motivation behind pre-processing is to take out aimpressive example from the background. For the most part, noise filtering, smoothing and standardization are to be done in this stage. The pre-processing additionally characterizes a smaller portrayal of the example. In this step we reshape our data into 28x28 matrix in order to use it in our model. Binarization changes over a gray scale image into a binary image.

* 1. **Segmentaion:**

Once the pre-processing step of the input images is done, sub-images of individual digits are formed from the sequence of images. Pre-processed digit images are segmented into a sub-image of individual digits, which are assigned a number to each digit. Each individual digit is resized into pixels. In this step an edge detection technique is being used for segmentation of dataset images.

**5.4 Feature Extraction:**

After completing of pre-processing stage and segmentation stage, the pre-processed images are represented in the form of a matrix which contains pixels of the images that are of very large size. In this way it will be valuable to represent the digits in the images which contain the useful information. This activity is called feature extraction. In the feature extraction stage redundancy from the data is eliminated.

14

**5.5 Creating the Model:**

Now we will **build our CNN model** in Python data science project. A CNN model generally consists of convolutional and pooling layers as discussed earlier. We have developed those different layers in our project. It works better for data that are represented as grid structures, this is the reason why CNN works well for image classification problems. The dropout layer is used to deactivate some of the neurons and while training, it reduces over fitting of the model. We will then compile the model with the Adam optimizer. In the last layer we used “softmax” function which is used in multi class classifier problem. As we will analyse digits from 0 to 9, so there are total of ten classes.

The**model.fit() function** of Keras will start the training of the model. It **takes the training data, validation data, epochs, and batch size. Epoch is one complete pass of the training dataset.**

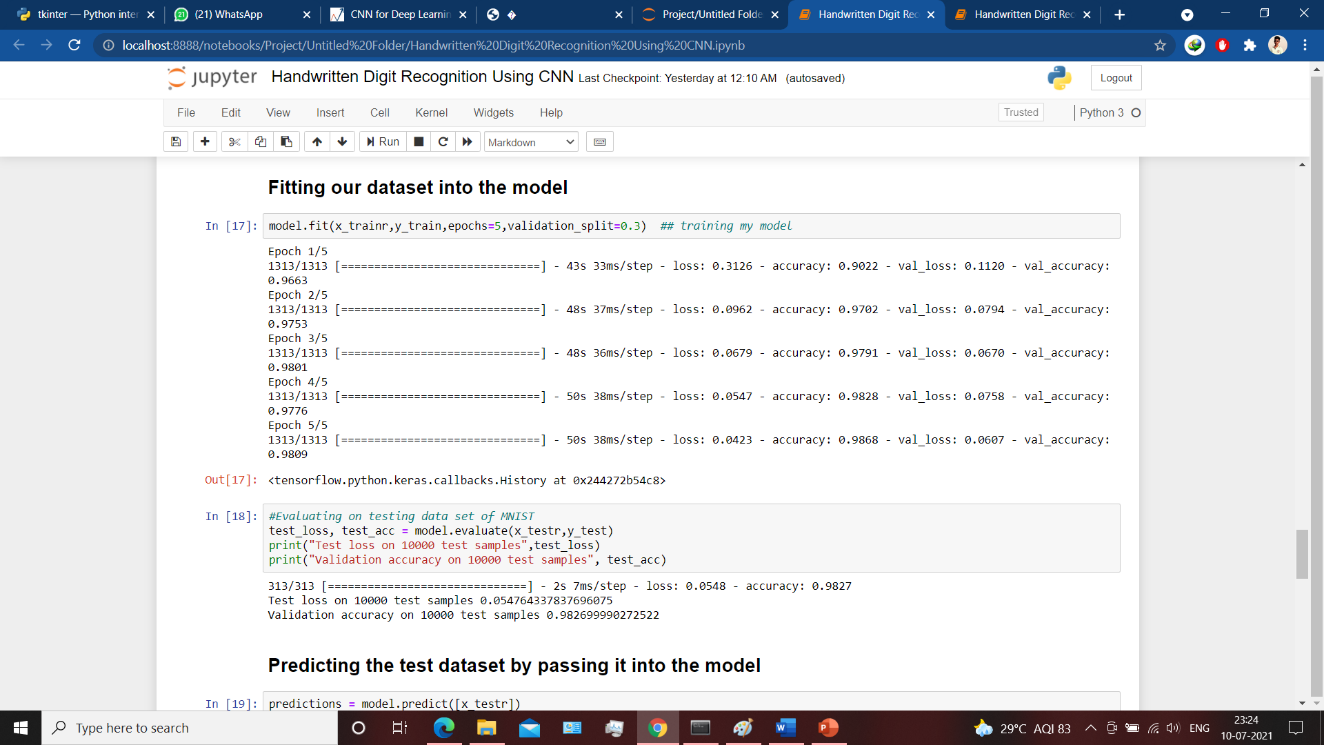


Figure 5.1: Training the dataset

It takes some time to train the model. After training, we save the weights and model definition as ‘hdr.h5’ file. This file will be used in the next part to implement the GUI.

We have 10,000 images in our test dataset which will be used to**evaluate how good our model performs.** The testing data was not involved in the training of the data therefore, it is new dataset for our model. The MNIST dataset is well balanced so we can get around 98% accuracy on our test dataset.

After creating the model we have predicted some data from our test dataset and show them by using matplotlib to know the accuracy of our model.

Below is the snapshot of that prediction. As we can see that our model is predicting the digit accurately from our test detaset.

15

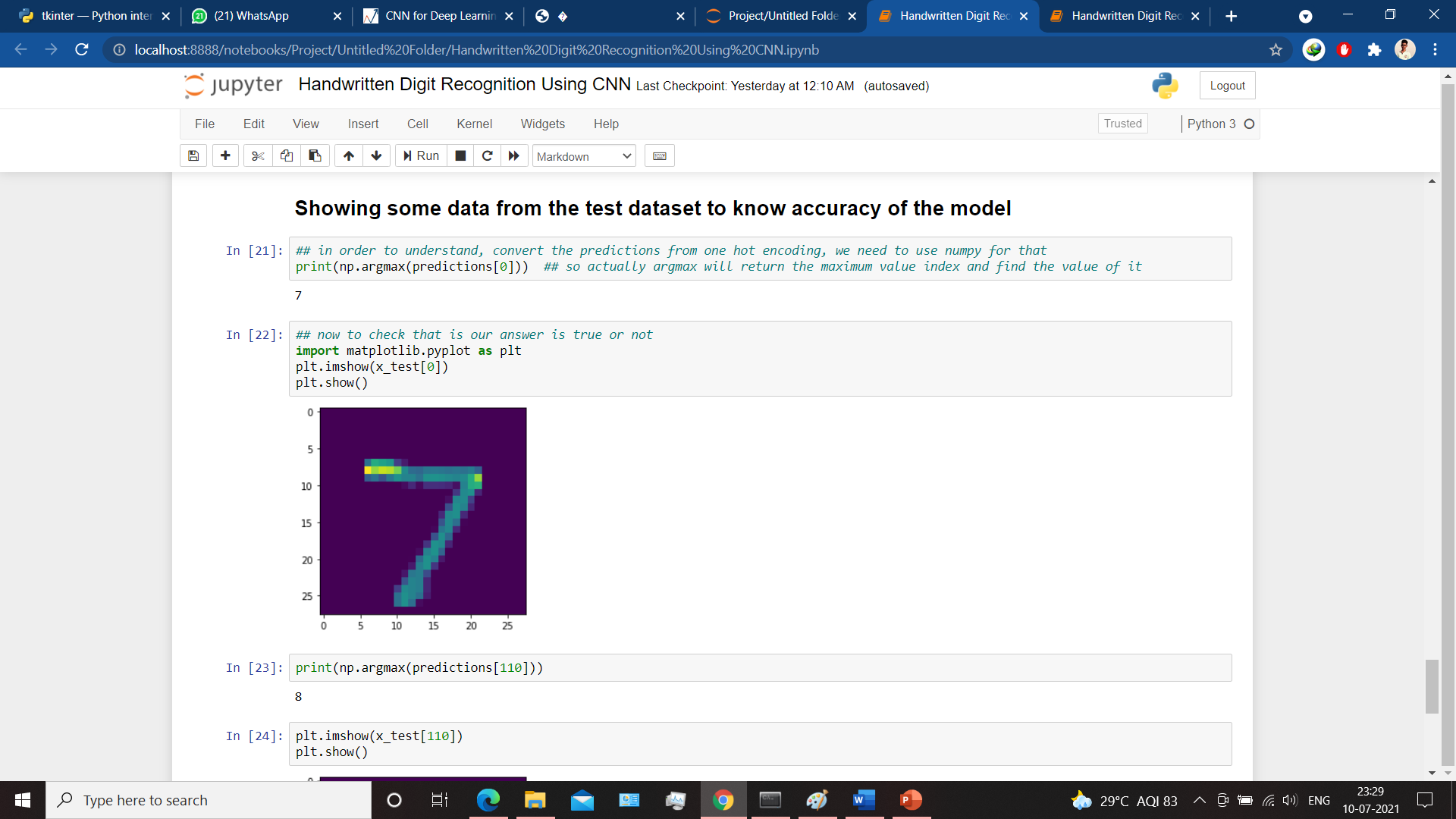


Figure 5.2: Prediction from test dataset

**5.6 Creating the GUI:**

Now after creating the model now is the time to build the Graphical User Interface (GUI) platform. We have used Tkinter library to build the GUI. It is very useful toolkit in python programming language and very easy to use. After importing required libraries we have initialized the GUI with the width and height and put in the background colour.

After initialization we have loaded our previously trained and saved model into the GUI by using keras.load\_model() function in order to use it predicting the input image drawn by the user.

Now in the GUI canvas we have put two buttons and define them. User will use those buttons to do their recognization task by pressing them.

**Open Paint:**

This is the button to open the paint application. When user will click on this button the paint application in our machine will automatically open where user can draw a digit by selecting the brush. Before drawing, the application should be set on the top left corner in order to grab the image part.

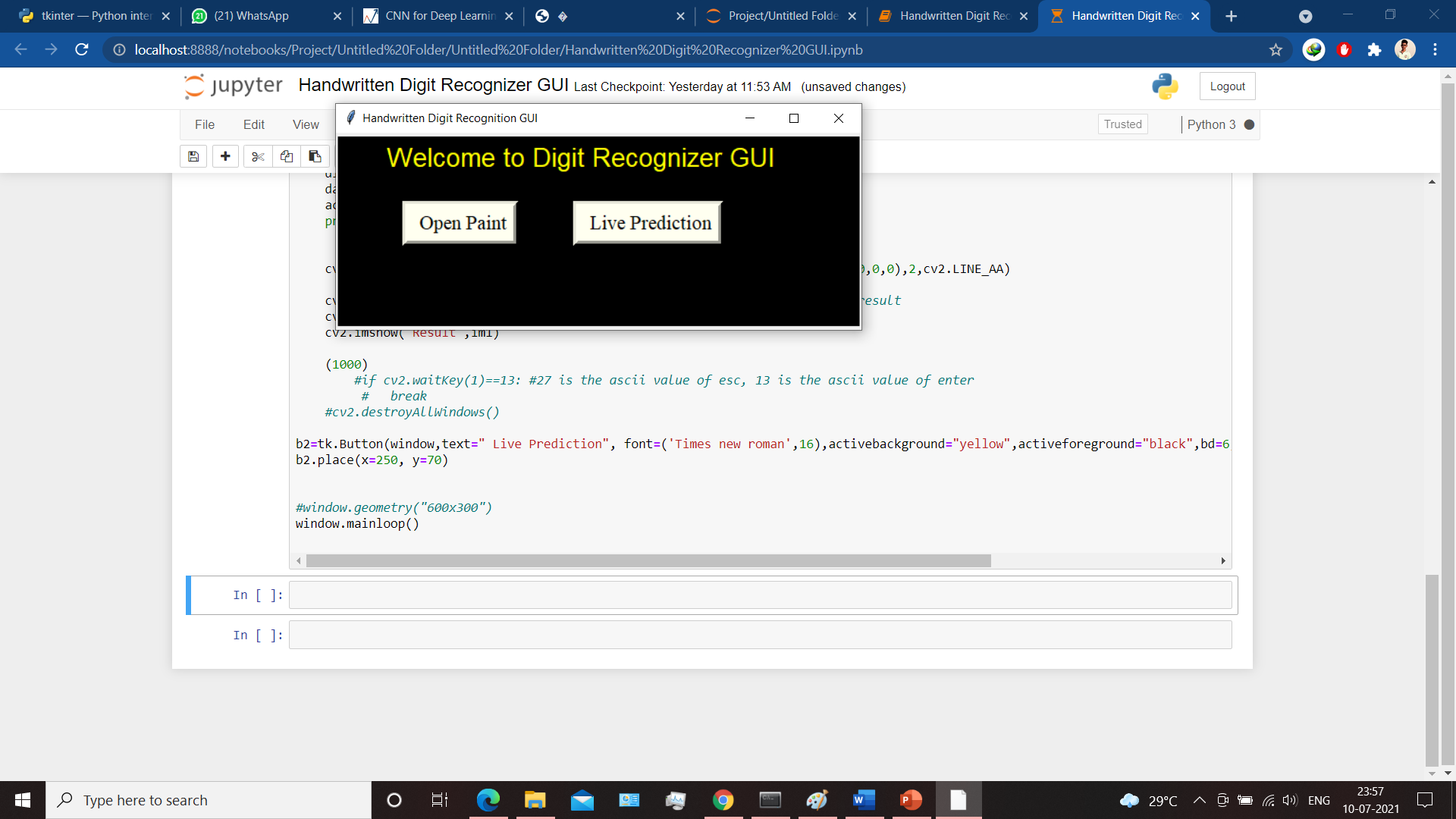
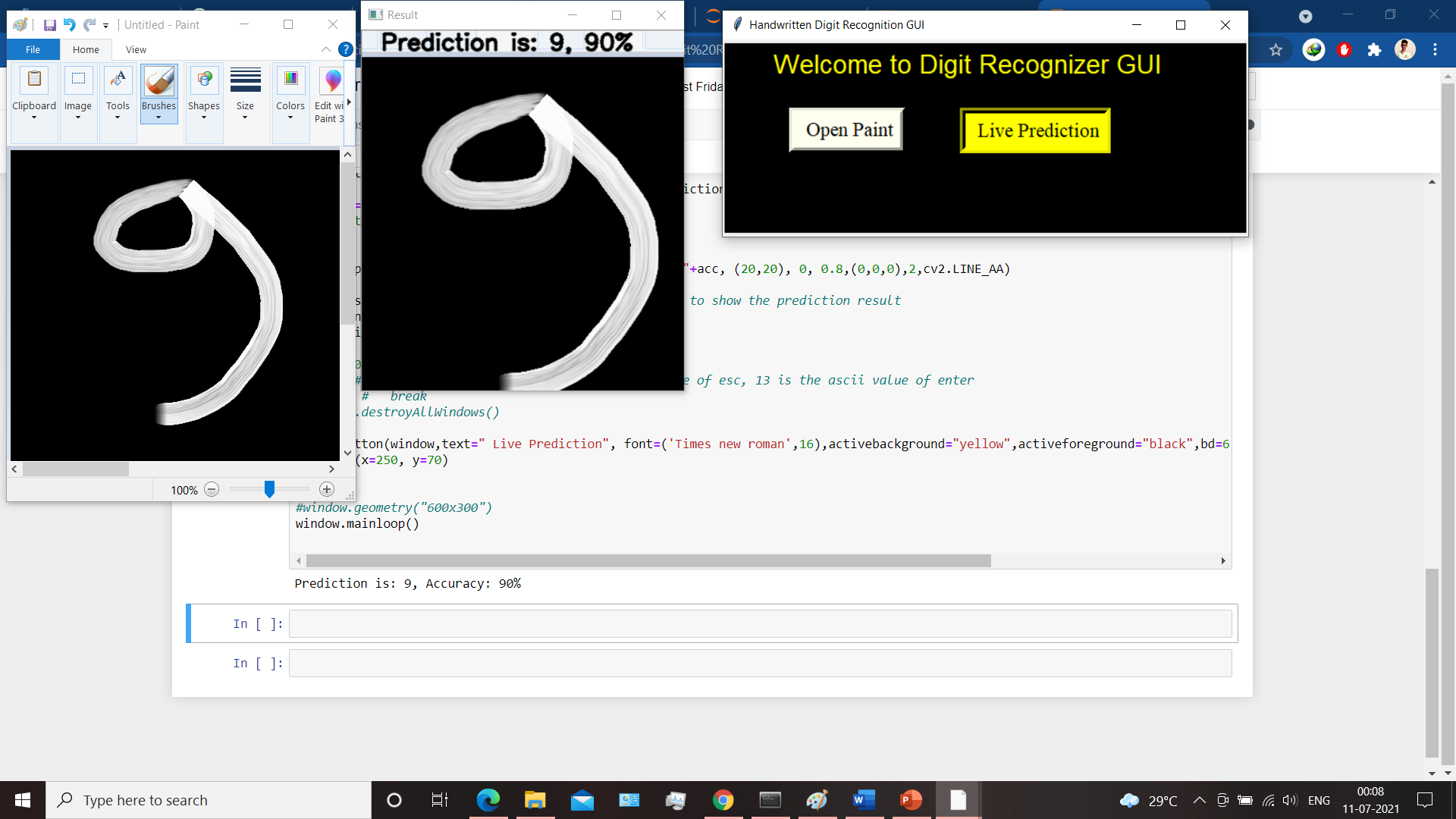


Figure 5.3: Front-end Design of the GUI

Above is the front end design of our GUI canvas. As we can see there are two buttons available.

16

**Live Prediction:**

This button is for prediction the image. After drawing the image if user press this button it will automatically capture the screen and grab the image. Then the pre-processing stage like the conversion to the grayscale, resized the image will be done. And after that the model will predict the image. After predicting it will show the result and the accuracy on another window. Here is an image that result window.

Figure 5.4: Prediction Result

**5.7 Result:**

After predicting the image the result will show on another window and the accuracy rate will also be shown there. So Finally we are able to recognize our own handwritten digits. Below is the screenshot of our whole running project.

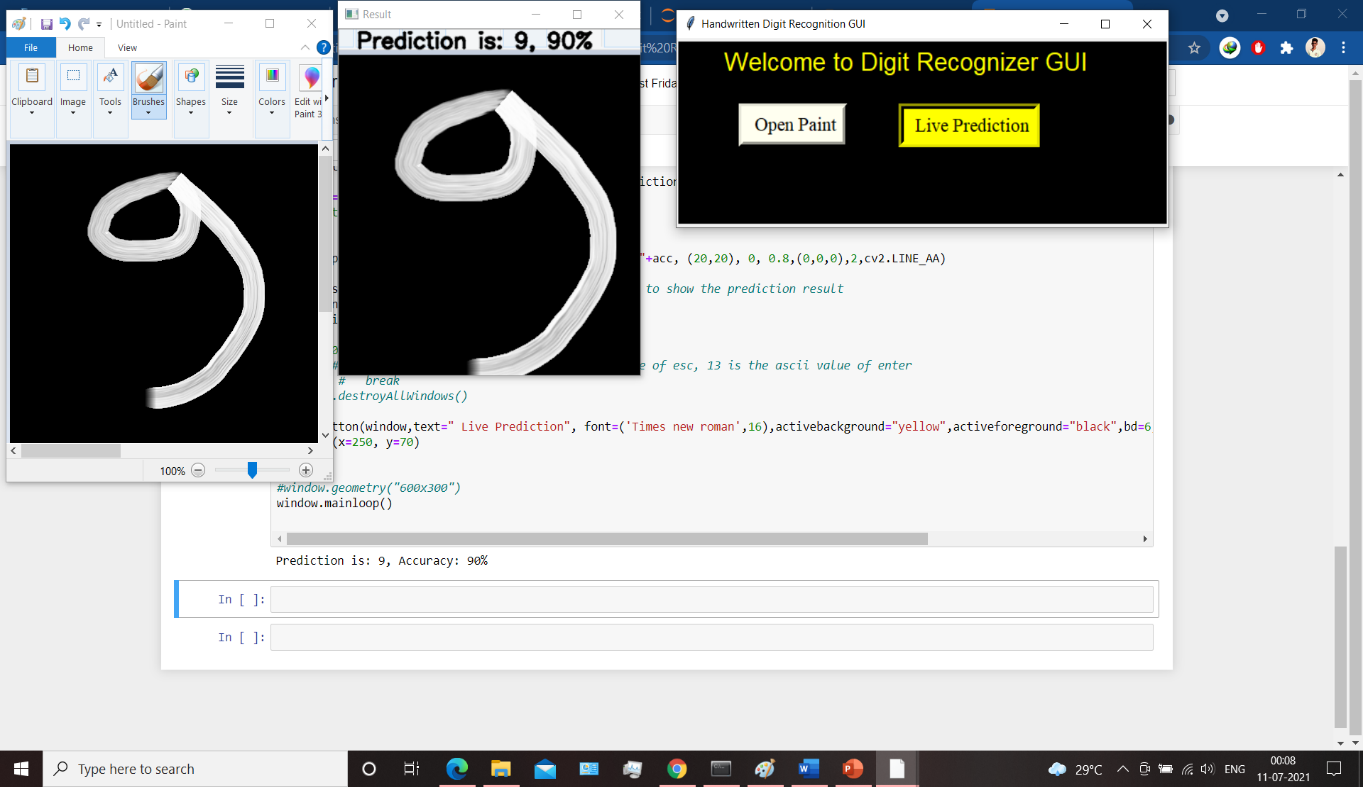


Figure 5.5: Running the whole project

So as we can see that we have achieved our goal and are able to predict our own handwritten digit. The predicted result is shown on another “Result” window and the accuracy rate is also be obtained. Our CNN model is giving 90% accuracy on our own drawn digit. Though it will vary from digit to digit according to their shapes, sizes and positions. Now we can predict the digit repeatedly by using the GUI

17

**LIMITATIONS:**

Dealing with a huge collection of digit images is not so easy task at all. So any machine system can’t give 100% accuracy for every individuals. As we can see our model is giving 90% accuracy in a certain digit. It changes from digit to digit based upon the the drawing. And after that one particular digit can be drawn in multiple ways. Below is a example of same handwritten digit drawn on paint in different way to understand it more deeply.

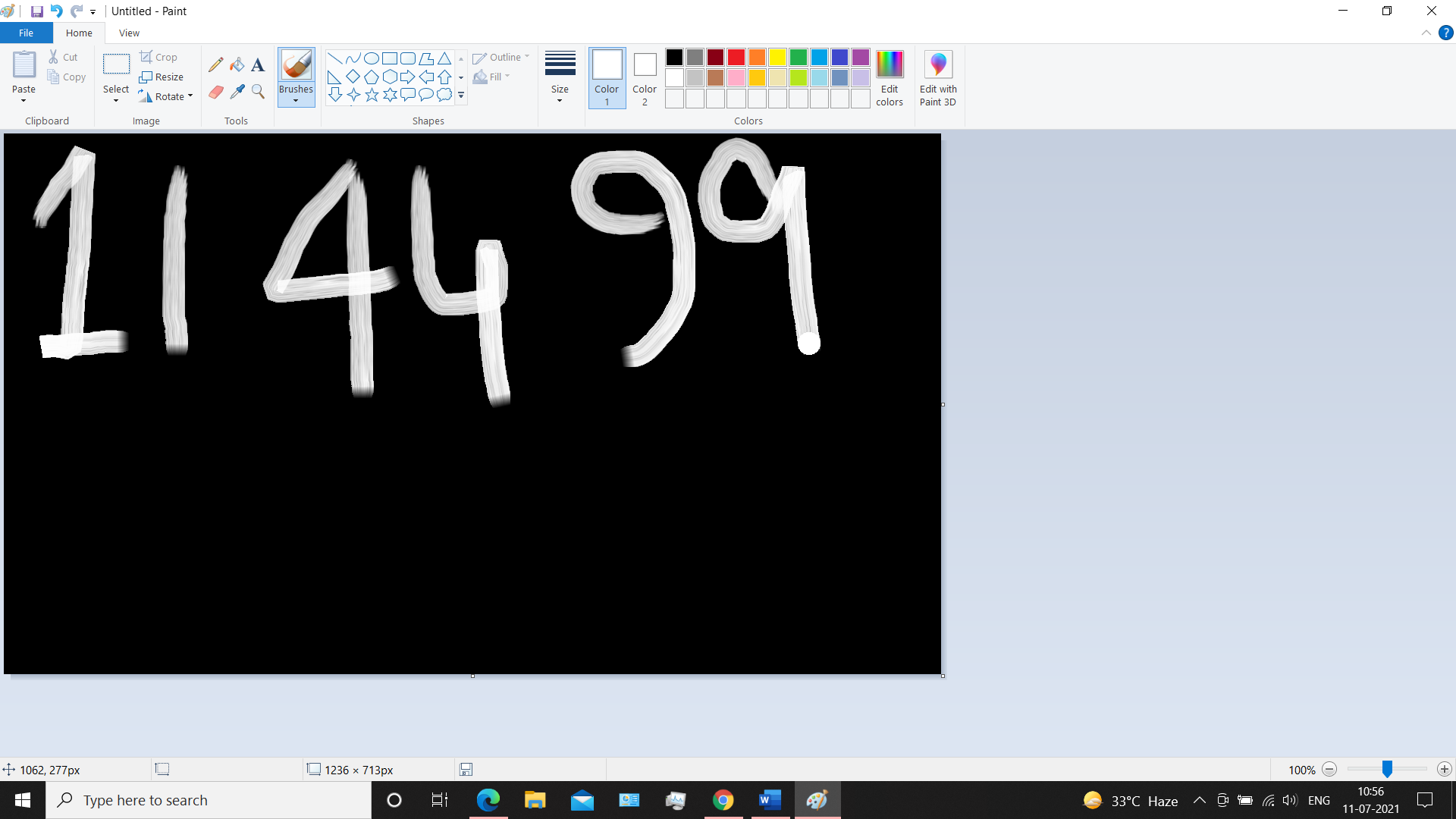


Figure 5.5: Different way of writing the same digit

So as we can see the same digit can be drawn in multiple ways. So in order to increase the accuracy we need to consider all these type of drawings. Moreover a digit drawn by a elder person and a child are different in terms of shapes, sizes, positions etc. So we need to consider all these things to recognize it more accurately. Besides these there are similarties in between the different digit. Like there is similarity between 1 and 7 in terms of shapes. So sometimes the model could be confused between these two.

We should also consider the lerarning issues that occur while training like overfitting amd underfitting and need to take care of these. However the accuracy can be increased by adding more dataset to the training and providing more number of layers in the model. So that this model and the concept can be used in different field in the real world like postal mail sorting, processing bank checks amounts, numeric entries in the form from paper documents etc. Moreover this concept can be used in the extended part of this field like character recognition, pattern recognition and so on.

18

**CONCLUSION AND FUTURE WORK**

Here we have developed a model which can identify handwritten digits drawn by our own. Later it can be extended for character recognition and real-time person’s handwriting. Handwritten digit recognition is the first step to the vast field of Artificial Intelligence and Computer Vision. As seen from the results of the experiment, CNN proves to its ability in predicting the digits accurately. The results can be made more accurate with more convolution layers and more number of hidden neurons and adding more number of dataset.

By using this deep learning technique, a high amount of accuracy can be obtained. Compared to other research methods, this method focuses on recognizing the handwritten digit by our classifier and it gives the accuracy almost 99% on the testing dataset. This model can also be developed by using other algorithms like support vector classifier (SVC), K- nearest neighour (KNN) and so on.

It can completely invalidate the need for manual typing. So there will be no need of investing money to the people for the labor of manual typing. Digit recognition is an excellent prototype problem for learning about neural networks and it gives a great way to develop more advanced techniques of deep learning.

We also have developed a GUI, by which an user can draw the digit he wants to predict by clicking the “Live Prediction” button. So for that they don’t need to understand the internal implementation of the code. This GUI concept could also be used in several fields like video capturing, website developing and so on.

Moreover this is just a part of artificial intelligence. So a lot of work in this field like face recognition, character recognition, and many other deep neural architecture project could be done using this concept.

**REFERENCES:**

19

14

1. [1]. M. Wu and Z. Zhang, Handwritten Digit Classification using the MNIST Dataset, 2010.
2. . A. Dutta and A. Dutta, Handwritten digit recognition using deep learning, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 7, July 2017.
3. Nimisha Jain, Kumar Rahul, IpshitaKhamaru. AnishKumar Jha, Anupam Ghosh (2017). “Hand Written Digit Recognition using Convolutional Neural Network (CNN)”, International Journal of Innovations & Advancement in Computer Science, IJIACS,ISSN 2347 – 8616,Volume 6, Issue 5.
4. Gaurav Jain, Jason Ko, Handwritten DigitsRecognition, Project Report, University of Toronto, 11/21/2008.
5. Saeed AL-Mansoori, "Intelligent Handwritten Digit Recognition using Artificial Neural Network", Int. Journal of Engineering Research and Applications, vol. 5, no. 5, pp. 46-51, 2015
6. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998), CrossRefGoogle Scholar
7. LeCun, Y., Cortes, C.: THE MNIST DATABASE of handwritten digits, http://yann.lecun.com/exdb/mnist/
8. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: Seventh International Conference on Document Analysis and Recognition, vol. 1, pp. 958–963 (2003), Google Scholar
9. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Deep, Big, Simple Neural Nets for Handwritten Digit Recogntion. Neural Computation 22(12), 3207–3220 (2010),
10. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Convolutional Neural Network Committees for Handwritten Character Classification. In: International Conference on Document Analysis and Recognition, vol. 10, pp. 1135–1139 (2011)
11. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column Deep Neural Networks for Image Classification. In: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, pp. 3642–3649 (2012)
12. Meier, U., Ciresan, D.C., Gambardella, L.M., Schmidhuber, J.: Better Digit Recognition with a Committee of Simple Neural Nets. In: International Conference on Document Analysis and Recognition, vol. 1, pp. 1250–1254 (2011)
13. Ranzato, A.M., Poultney, C., Chopra, S., LeCun, Y.: Efficient Learning of Sparse Representations with an Energy-Based Model. In: Advances in Neural Information Processing Systems 19, NIPS 2006 (2006)
14. Cruz, R., Cavalcanti, G., Ren, T.: Handwritten digit recognition using multiple feature extraction techniques and classifier ensemble. In: International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 215–218 (2010)
15. Sesmero, M.P., Alonso-Weber, J.M., Gutiérrez, G., Ledezma, A., Sanchis, A.: A new artificial neural network ensemble based on feature selection and class recoding. Neural Computing and Applications 21(4), 771–783 (2010)

20

1. Alonso-Weber, J.M., Sanchis, A.: A Skeletonizing Reconfigurable Self-Organizing Model: Validation Through Text Recognition. Neural Processing Letters 34(1), 39–58 (2011)
2. Matsuoka, K.: Noise Injection into Inputs in back-propagation learning. IEEE Transactions on Systems, Man, and Cybernetics 22(3), 436–440 (1992)
3. An, G.: The Effects of Adding Noise During Backpropagation Training on a Generalization Performance. Neural Computation 674, 643–674 (1996)
4. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research 11(1), 169–198 (1999)
5. Dietterich, T.: Ensemble methods in machine learning. In: Multiple Classifier Systems, pp. 1–15 (2000)
6. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: a survey and categorisation. Information Fusion 6(1), 5–20 (2005)
7. Perrone, M.P., Cooper, L.N.: When networks disagree: Ensemble methods for hybrid neural networks. In: Mammone, R.J. (ed.) Neural Networks for Speech and Image Processing, ch. 10. Chapman-Hall (1993).

21